

Σημασιολογικός Ιστός και Ευφυείς Εφαρμογές

Μέλη Ομάδας:

- Σωτήριος Λουκάς Καμπύλης, ΑΕΜ: 3805

- Ραφαήλ Τανακίδης, ΑΕΜ: 3814

Η συγκεκριμένη εργασία υλοποιήθηκε από τα 2 παραπάνω μέλη και αναπαριστά την μοντελοποίηση ενός κοινού θεματικού πεδίου ενδιαφέροντος, δηλαδή μιας **Αλυσίδας Βιβλιοθήκης-Βιβλιοπωλείου**. Η εργασία υλοποιήθηκε εξίσου και από τα 2 μέλη, καθώς βρισκόμασταν σε επικοινωνία από την αρχή της υλοποίησης της μέχρι το τέλος. Μάλιστα, πριν ξεκινήσουμε την *σχεδίαση/μοντελοποίηση* της οντολογίας, καταγράψαμε σχετικές έννοιες που θεωρήσαμε θα φανούν χρήσιμες και μετά αρχίσαμε να τις τοποθετούμε στο *TopBraid Composer*! Ύστερα, τοποθετήσαμε τις Κλάσεις (21/25), δυστυχώς δεν μπορέσαμε να βρούμε παραπάνω έτσι ώστε να λειτουργούν όλα σωστά, τις υπό-κλάσεις (13/10) και προσπαθήσαμε να διατηρήσουμε την σωστή ιεραρχία ανάμεσα σε αυτές (και σύμφωνα με τον *Reasoner*). Έπειτα, τοποθετήσαμε (22/15) ιδιότητες που παίρνουν τιμές άλλα αντικείμενα, τοποθετήσαμε (29/15) ιδιότητες που παίρνουν τιμές άλλα αντικείμενα και (3/5) υπό-ιδιότητες, δυστυχώς ούτε εδώ δεν μπορέσαμε να βρούμε παραπάνω έτσι ώστε να λειτουργούν όλα σωστά. Μάλιστα, δόθηκε ιδιαίτερη προσοχή στα *Domain/Ranges* του κάθε πεδίου (και γίναν αρκετές δοκιμές με τον *Reasoner*) έτσι ώστε να μπορεί να εξάγει λογικά συμπεράσματα. Τέλος, δημιουργήθηκαν (86/50) Αντικείμενα, δώσαμε τιμές σε όλες τις σχετικές ιδιότητες κάθε αντικειμένου και εκτελέσαμε 3 συγκεκριμένα ερωτήματα μέσω του SPARQL. Παρακάτω θα αναλύσουμε περαιτέρω την σημασία των κλάσεων/ιδιοτήτων για το μοντέλο μας (έτσι ώστε να γίνει καλύτερα αντιληπτό). Ωστόσο, δεν θα αναλύσουμε εξονυχιστικά όλο το μοντέλο γιατί χρησιμοποιήσαμε πάρα πολλά πράγματα και θα μας έπαιρνε τουλάχιστον 10 σελίδες (μόνο για αυτό), οπότε θα αναλύσουμε ένα μεγάλο δείγμα από το μοντέλο μας.

ΚΛΑΣΕΙΣ/ΥΠΟ-ΚΛΑΣΕΙΣ

Κλάσεις και Υπό-κλάσεις που χρησιμοποιήθηκαν (με την σειρά που εμφανίζονται στο Top Braid Composer): 1) Action, 2) Borrow, 3) Buy, 4) Read, 5) BorrowingCard, 6) ClubJoin, 7) Clubs, 8) Communication, 9) Donation, 10) HasBorrowingCard, 11) Location, 12) Organization, 13) Person, 14) Author, 15) Client, 16) Stuff, 17) Product, 18) Book, 19) Programs, 20) AfterSchoolHomeWorkHelp, 21) Games, 22) Publisher, 23) Receipt, 24) Salary, 25) Schedule, 26) Security, 27) Camera, 28) Guard, 29) Store, 30) Supplier, 31) Technology, 32) Pc, 33) Printer & 34) Warehouse.

Αν και οι περισσότερες Κλάσεις & Υπό-κλάσεις έχουν σχεδιαστεί για να είναι *αυτοπεριγραφικές* παρακάτω δίνω μια εξήγηση για μερικές (που κρίνονται *δυσνόητες*):

1) Action = Ενέργεια που θα πραγματοποιήσει ο 15) Client στην Βιβλιοθήκη/Βιβλιοπωλείο μας.

5) BorrowingCard = Δανειστική κάρτα του 15) Client για την περίπτωση της Βιβλιοθήκης.

7) Clubs = Κλαμπ που μπορεί να ενταχθεί ως μέλος ο 15) Client μέσω του 6) ClubJoin.

11) Location = Τοποθεσία της Βιβλιοθήκης-Βιβλιοπωλείο μας (αυτή η κλάση κρίνεται εξαιρετικά σημαντική λόγω του ότι αναφερόμαστε σε <<**Αλυσίδα Βιβλιοθηκών/Βιβλιοπωλείων**>>).

12) Organization = Οργάνωση που μπορεί να δραστηριοποιείται εθελοντικά σε κάποια από τις Βιβλιοθήκες/Βιβλιοπωλεία μας.

14) Author = Συγγραφέας κάποιου 18) Book.

19) Programs = Ενεργά προγράμματα που δραστηριοποιούνται σε κάποια από τις Βιβλιοθήκες/Βιβλιοπωλεία μας.

24) Salary = Μισθός ενός 16) Stuff σε κάποια από τις Βιβλιοθήκες/Βιβλιοπωλεία μας.

25) Schedule = Ωρολόγιο πρόγραμμα κάποιας από τις Βιβλιοθήκες/Βιβλιοπωλεία μας.

26) Security = Η ασφάλεια σε κάποια από τις Βιβλιοθήκες/Βιβλιοπωλεία μας.

ΙΔΙΟΤΗΤΕΣ/ΥΠΟ-ΙΔΙΟΤΗΤΕΣ

Ιδιότητες και Υπό-ιδιότητες που χρησιμοποιήθηκαν (με την σειρά που εμφανίζονται στο Top Braid Composer): 1) ActionClient, 2) BookAuthor, 3) BookCategory, 4) BookPublisher, 5) BookReleaseDate, 6) BookTitle, 7) BorrowBook, 8) BorrowEnd, 9) BorrowingCardId, 10) BorrowingCardUntil, 11) BorrowStart, 12) BorrowStuff, 13) BuyProduct, 14) BuyReceipt, 15) BuyStuff, 16) CameraId, 17) ClubJoinClub, 18) ClubJoinPerson, 19) ClubType, 20) CommunicationEmail, 21) CommunicationPhone, 22) DonationAmount, 23) DonationPerson, 24) GuardIs, 25) HasBorrowingCardCard, 26) HasBorrowingCardClient, 27) LocationAddress, 28) LocationCity, 29) LocationZipCode, 30) OrganizationArrive, 31) OrganizationContent, 32) OrganizationLeave, 33) PersonAge, 34) PersonId, 35) PersonName, 36) ProductName, 37) ProductValue, 38) PublisherId, 39) PublisherLocation, 40) PublisherName, 41) ReadBook, 42) SalaryAmount, 43) SalaryStuff, 44) ScheduleStuff, 45) ScheduleTime, 46) StoreCommunication, 47) StoreLocation, 48) SupplierName, 49) SupplierType, 50) SupplierTypeQuantity, 51) TechnologyAvailable, 52) WarehouseCommunication & 53) WarehouseLocation.

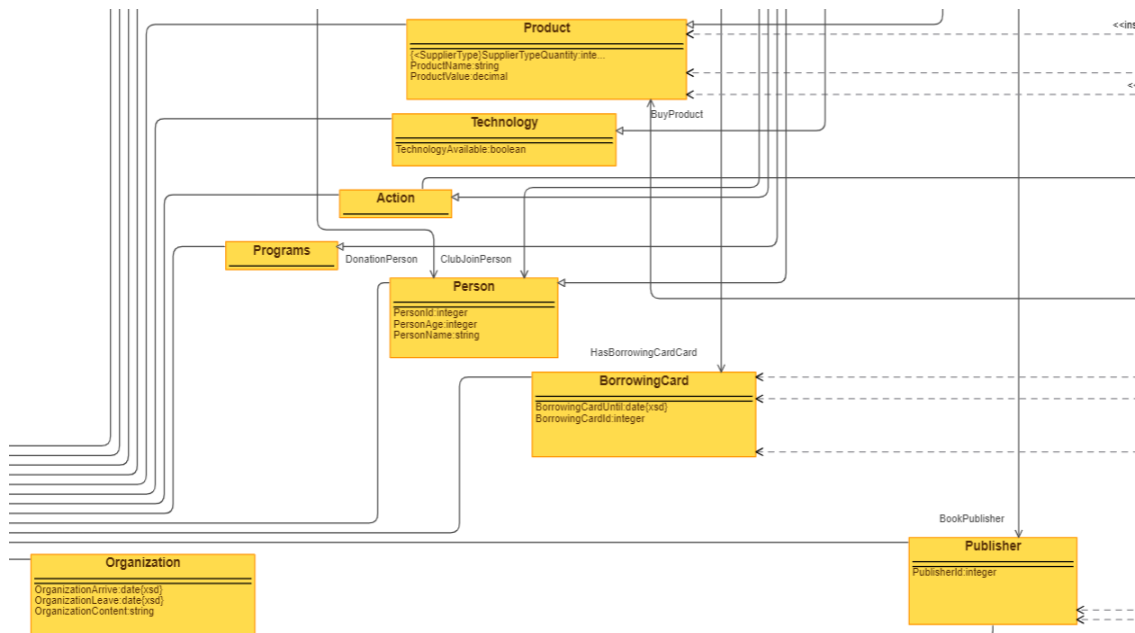
Αν και οι περισσότερες Ιδιότητες & Υπό-Ιδιότητες έχουν σχεδιαστεί για να είναι *αυτοπεριγραφικές* παρακάτω δίνω μια εξήγηση για μερικές (που κρίνονται *δυσνόητες*), όπως και στις Κλάσεις/Υπό-Κλάσεις:

1) ActionClient = Η ιδιότητα αυτή αναπαριστά την ενέργεια που θα κάνει ο πελάτης μέσα στην Βιβλιοθήκη/Βιβλιοπωλείο μας.

24) GuardIs = Η ιδιότητα αυτή δείχνει ποιος είναι ο φύλακας μέσα σε κάποια από τις Βιβλιοθήκες/Βιβλιοπωλεία μας.

30) OrganizationArrive = Η ιδιότητα αυτή μας βοηθάει να καταλάβουμε ποια στιγμή έφτασε κάποια οργάνωση που μπορεί να δραστηριοποιείται εθελοντικά σε κάποια από τις Βιβλιοθήκες/Βιβλιοπωλεία μας.

50) SupplierTypeQuantity = Η ιδιότητα αυτή μας παρουσιάζει την ποιότητα των βιβλίων που μας στέλνει ένας εκδότης. Μάλιστα, είναι υπό-ιδιότητα της 49) SupplierType!



Αναπαράσταση Μερικών Κλάσεων από το Μοντέλο μας (μέσω OWLGrE)

SPARQL

1^ο Ερώτημα: Ποια Βιβλία του Συγγραφέα author_1 έχουν νοικιαστεί από τον client_1;

```

SELECT ?book
WHERE {
  ?book Library:BookAuthor Library:Author_1 .
  ?borrow Library:BorrowBook ?book .
  ?borrow Library>ActionClient Library:Client_1 .
}
  
```

Results:

- [book]
- Library:Book_1
- Library:Book_2

2^ο Ερώτημα: Ποιο είναι το τηλέφωνο της αποθήκης στην Αθήνα;

```

SELECT ?phone
WHERE {
  ?warehouse Library:WarehouseLocation ?location .
  ?location Library:LocationCity "Athens" .
  ?warehouse Library:WarehouseCommunication ?communication .
  ?communication Library:CommunicationPhone ?phone .
}
  
```

Results:

- [phone]
- phone 1

3^ο Ερώτημα: Ποιοι πελάτες συμμετέχουν στο Club: "Thoughts" & έχουν δανειστική κάρτα;

The screenshot shows a SPARQL query editor interface. The top bar includes tabs for Imports, Instances, Domain, Error Log, SPARQL, Text Search, and Inferences. The main area is divided into a Query Editor and a Query Library. The Query Editor contains the following SPARQL query:

```
SELECT ?people
WHERE {
  ?temp Library:ClubType "Thoughts" .
  ?clubs Library:ClubJoinClub ?temp .
  ?clubs Library:ClubJoinPerson ?people .
  ?card Library:HasBorrowingCardClient ?people
}
```

The Query Library on the right shows the results of the query, which are two instances: Library:Client_1 and Library:Client_2. The results are displayed in a table with a header row [people] and two data rows.

[people]
Library:Client_1
Library:Client_2